Classificação Local utilizando Least Squares Support Vector Machine (LSSVM)

Rômulo B. P. Drumond, Renan F. Albuquerque, Diego P. Sousa, Guilherme A. Barreto
Programa de Pós-Graduação em Engenharia de Teleinformática (PPGETI)
Universidade Federal do Ceará (UFC) - Campus do Pici
Av. Mister Hull, S/N - Bloco 725, Centro de Tecnologia
CP 6005, CEP 60455-970, Fortaleza, Ceará
romulo.drumond@alu.ufc.br, {renanfonteles, diegoperdigao}@gmail.com, gbarreto@ufc.br

***Resumo*—Os modelos de classificação global são métodos que utilizam todo o conjunto de dados de treinamento disponível para a construção de um único modelo que especifique a superfície de separação dos dados. Alternativamente, modelos de classificação local baseiam-se na construção de classificadores locais treinados a partir de subconjuntos dos dados de treinamento. Este artigo apresenta um estudo sobre a abordagem de classificação local para projeto de classificadores baseados em máquinas de vetores-suporte de mínimos quadrados (LSSVM). As partições locais foram definidas a partir do algoritmo de agrupamento K-médias. Os dados dos agrupamentos resultantes foram utilizados para treinar cada modelo LSSVM local. Diversos índices de validação de agrupamentos foram utilizados como critério de determinação do número de partições locais para cada problema de classificação estudado. Experimentos com vários conjuntos de dados de classificação foram realizados para comparar a abordagem local com a global.**

***Keywords*—Reconhecimento de Padrões; Modelos de Classificação Local; Máquinas de Vetores Suporte; K-médias; Least Squares Support Vector Machine;**

I. Introdução

Algoritmos de classificação são modelos que definem superfícies de decisão capazes de separar dados de diferentes classes. O modelo de classificação, ou classificador, deve fornecer uma função que mapeie todos os dados pertencentes ao espaço de entrada para um espaço de saída composto por valores discretos (i.e., rótulos) que representam as classes do problema. Os problemas de classificação podem ser categorizados em linearmente separáveis e não linearmente separáveis. Na prática, é muito difícil inferir a priori a complexidade de um certo problema de classificação (ou seja, se este é linearmente separável ou não) apenas a partir do conjunto de dados. Assim, adota-se um postura comparativa em que diferentes classificadores de diferentes complexidades são testados e escolhe-se aquele em que apresentar melhor acurácia com menor custo computacional.

Existem diversos paradigmas de aprendizado que são utilizados como estratégias na construção de modelos de classificação. Algoritmos baseados em redes neurais artificiais (ANN) e máquinas de vetores-suporte (SVM) são exemplos de abordagens de classificação amplamente exploradas na solução de problemas complexos de classificação. Essas técnicas são geralmente utilizadas com base na abordagem de modelagem global. A modelagem global consiste em projetar um único modelo que represente os dados disponíveis no conjunto de treino, constituídos de observações de entrada e saída. Em contraste com essa abordagem, a modelagem local é baseada na segmentação do espaço de entrada em várias partições menores, onde para cada partição é construído um modelo específico. A modelagem local é uma alternativa para problemas não lineares, como exemplo na área de regressão [1]–[3]. No estado da arte em classificação de dados, vários estudos exploram métodos e abordagens envolvendo estratégias baseadas em modelagem local. Modelos locais já foram apresentados em artigos clássicos na área de redes neurais artificiais, como perceptrons simples locais [4] e misturas de especialistas lineares locais [5]. Mais recentemente, vários artigos propuseram métodos de discriminação local na construção de classificadores [6]–[8]. Um algoritmo de uso bastante difundido na literatura de aprendizado de máquinas é o classificador baseado em vetores-suporte de mínimos quadrados (LSSVM, *least squares support vector machine*). O classificador LSSVM, que pode ser entendido como uma variante não-esparsa do algoritmo SVM de Cortes & Vapnik [9], tem sido aplicado com sucesso em tarefas diversas [10], incluindo predição [11] e classificação [12]. Entretanto, apesar do bom desempenho em tais tarefas, o uso do classificador LSSVM em estratégias de classificação local não tem sido observado.

Neste trabalho, apresentamos um modelo de classificação local que consiste na combinação do algoritmo K-médias com o classificador LSSVM. O algoritmo de agrupamento K-médias é utilizado como estratégia de definição de agrupamentos locais (i.e., partições) nos dados de treinamento. As partições são então interpretadas como subproblemas distintos de classificação, onde a cada partição será associado um classificador LSSVM treinado localmente. O restante do artigo está organizado da seguinte forma. A seção 2 apresenta os fundamentos de classificação local e descreve a estratégia utilizada na determinação das partições locais. Esta baseia-se na utilização do algoritmo de agrupamento K-médias, e utiliza índices de validação de agrupamentos para reduzir o espaço de busca do hiper-parâmetro K, ou número de partições. Na seção 3, os fundamentos do classificador LSSVM são descritos brevemente. A seção 4 contém os experimentos realizados, apresentando os conjuntos de dados, a metodologia e as

métricas de desempenho utilizadas no estudo da abordagem proposta. Na seção 5 são discutidos os resultados experimentais. Por fim, a seção 6 conclui este artigo com uma discussão sobre as vantagens e desvantagens do classificador LSSVM local e apresenta sugestões de trabalhos futuros.

II. Definição das partições utilizando K-médias

A. Classificação Local

Os modelos de classificação local consistem em métodos de aprendizado que buscam uma função discriminante $g(\cdot)$ para um subconjunto específico (ou partição). Considere um conjunto de treinamento $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, $\forall \mathbf{x}_i \in \mathbb{R}^d$, em que N é o número de amostras de treinamento e d é a dimensionalidade do vetor de entrada. O conjunto de treinamento $\mathcal{X}$ será separado em um conjunto de partições disjuntas $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$, onde K é o número de partições. Portanto, $\forall V_i, V_j \in \mathcal{V}$, com $i \neq j \implies V_i \bigcap V_j = \emptyset$. Para cada partição V_i, um respectivo modelo M_i será construído. Na modelagem local, supõe-se que o espaço de entrada seja composto por várias partições de interesse. Normalmente, medidas de distância e similaridade são usadas para definir formalmente essas partições [2]. O paradigma da modelagem local pode ser descrito em três etapas: (i) determinação das partições (e.g. através do K-médias), (ii) treinar um modelo para cada partição (e.g., para cada partição de dados fornecida pelo K-médias) e (iii) na predição determinar a partição V_i de uma nova amostra baseado em algum critério de similaridade (e.g. distância euclidiana aos protótipos do K-médias) e usar o respectivo modelo M_i para a inferência da classe.

B. K-médias

K-médias é um algoritmo de agrupamento que separa o conjunto de dados em partições (ou clusters) em que cada amostra tende a pertencer ao cluster com a média mais próxima. O algoritmo pode ser resumido em cinco etapas principais:

1) Define-se K, que representa a quantidade de agrupamentos ou protótipos.
2) Inicializa-se a posição dos K protótipos. O i-ésimo protótipo é representado pelo vetor de pesos $\mathbf{w}_i$.
3) Divide-se o conjunto de dados em K partições (ou clusters). Cada partição V_i será definida como
$$V_i = \left\{\mathbf{x} \in \mathbb{R}^d \;\middle|\; \|\mathbf{x}-\mathbf{w}_i\|_2^2 < \|\mathbf{x}-\mathbf{w}_j\|_2^2,\ \forall j \neq i\right\}, \quad (1)$$
em que $\|\cdot\|_2$ denota a norma euclidiana.
4) Calcula-se a nova posição de $\mathbf{w}_i$ a partir da média aritmética das N_i amostras da partição V_i, utilizando a equação
$$\mathbf{w}_i = \frac{1}{N_i} \sum_{\forall \mathbf{x} \in V_i} \mathbf{x}. \quad (2)$$
5) Repete-se os passos 3 e 4 até a convergência.

O resultado do algoritmo K-médias é uma matriz $\mathbf{W} \in \mathbb{R}^{K \times d}$ contento os K vetores $\mathbf{w}_i$ que servirão para determinar as partições do modelo de classificação local. Uma prática comum para avaliar a convergência é através de métricas de dispersão dos protótipos, como a soma das distâncias quadradas (SSD, *sum of squared distances*), definida por

$$SSD(K) = \sum_{i=1}^{K} \sum_{\forall \mathbf{x} \in V_i} \|\mathbf{x} - \mathbf{w}_i\|_2^2 . \quad (3)$$

Dado que a convergência desse algoritmo depende fortemente de inicialização dos protótipos, é comum executá-lo diversas vezes e adotar o melhor resultado segundo alguma métrica (e.g., menor SSD) . Mais detalhes sobre o algoritmo K-médias podem ser encontrados em [13].

C. Índices de Validação de Agrupamentos

Neste trabalho, índices de validação de *clusters* foram utilizados na seleção do hiper-parâmetro K do algoritmo K-médias. Este hiper-parâmetro determina a quantidade de partições locais para o classificador local, alvo de estudo neste artigo. Os índices de validação utilizados neste trabalho foram: Calinski-Harabasz [14], Dunn [15], Davies-Bouldin [16] e Silhouette [17].

1) Índice Calinski-Harabasz (CH): Proposto por [14], o índice CH é modelado através da seguinte expressão

$$CH(K) = \frac{trace(\mathbf{B}_K)/(K-1)}{trace(\mathbf{W}_K)/(N-K)}, \quad (4)$$

onde $\mathbf{B}_K = \sum_{l=1}^{K} \#V^l (\mathbf{x}^l - \mathbf{x})(\mathbf{x}^l - \mathbf{x})'$ é a soma dos quadrados entre *clusters*, N é número total de amostras, K é o número de *clusters*, $\mathbf{W}_K = \sum_{l=1}^{K} \sum_{i=1}^{\#V^l - 1} \sum_{j=i+1}^{\#V^l} (\mathbf{x}i^l - \mathbf{x}j^l)(\mathbf{x}i^l - \mathbf{x}j^l)'$ é a soma dos quadrados *intracluster*, $\mathbf{x}i^l$ e $\mathbf{x}j^l$ são, respectivamente, os i-ésimos e j-ésimos itens do *cluster* V^l, $\mathbf{x}^l$ é o centróide do l-ésimo *cluster*, $\#V^l$ é a cardinalidade do l-ésimo *cluster* e $\mathbf{x}$ é o centróide do banco de dados inteiro.

A escolha da quantidade de *clusters* K é dada pelo argumento que maximiza o índice CH. Perceba que $CH(K)$ não pode ser usado para $K = 1$.

2) Família de Índices Dunn: A família de índices Dunn, proposta por [15], é representada genericamente pela expressão

$$D(K) = \frac{\min_{i \neq j}\{\delta(V_i, V_j)\}}{\max_{1 \leq l \leq K}\{\Delta(V_l)\}}, \quad (5)$$

onde $\delta(V_i, V_j)$ denota uma função de dissimilaridade (e.g., distância euclidiana) entre os agrupamentos V_i e V_j, e $\Delta(V_l)$ é uma medida da dispersão dos dados dentro do agrupamento V_l.

No índice Dunn original, $\delta(V_i, V_j)$ é definido pela equação:

$$\delta(V_i, V_j) = \min_{\mathbf{x} \in V_i,\ \mathbf{y} \in V_j} \{d(\mathbf{x}, \mathbf{y})\}, \quad (6)$$

e $\Delta(V_i)$ é dada pela equação

$$\Delta(V_i) = \max_{\mathbf{x}, \mathbf{y} \in V_i} \{d(\mathbf{x}, \mathbf{y})\}. \quad (7)$$

Para esta família de índices, o valor de K gerador do valor máximo de $D(K)$ deve ser escolhido o número ótimo de agrupamentos.

3) Índice Davies-Bouldin (DB): Proposto por [16], o índice DB é modelado a partir da razão entre a soma da dispersão dentro dos agrupamentos e da dispersão entre agrupamentos. Neste índice, a métrica $R_{i,j}$ representa a seguinte medida de similaridade entre os agrupamentos V_i e V_j:

$$R_{i,j} = \frac{e_{V_i} + e_{V_j}}{d(V_i, V_j)}, \tag{8}$$

em que e_{V_i} e e_{V_j} são, respectivamente, os erros médios para os agrupamentos V_i e V_j, e $d(V_i, V_j)$ denota a distância euclidiana entre os protótipos dos dois agrupamentos.

Seja o índice para o k-ésimo agrupamento dado indicado por:

$$R_k = \max_{i \neq k}\{R_{i,k}\}. \tag{9}$$

O índice de DB é representado pela expressão

$$DB(K) = \frac{1}{K}\sum_{k=1}^{K} R_k. \tag{10}$$

Por fim, o valor de K mais adequado deve apresentar o menor valor de $DB(K)$.

4) Silhuetas: Proposta por [17], o índice de silhuetas (Sil) é definido como

$$\begin{aligned} Sil(K) =& \sum_{i=1}^{N} S(i)/N, \\ S(i) =& [b(i) - a(i)]/\max\{a(i), b(i)\}, \end{aligned} \tag{11}$$

em que $a(i)$ denota a dissimilaridade média do i-ésimo vetor de características à todos outros vetores pertencentes ao mesmo agrupamento, já a variável $b(i)$ representa a menor dissimilaridade média deste i-ésimo vetor de características à qualquer outro agrupamento a qual ele não pertença. A métrica de silhuetas pode ser calculada a partir de qualquer medida de dissimilaridade, tais como as distâncias Euclidiana e Manhattan. O valor de K a ser definido como o número ótimo de agrupamentos é aquele que produz o maior valor de $Sil(K)$.

III. O Classificador LSSVM

O classificador LSSVM é uma variação do classificador SVM no qual pequenas mudanças na definição do problema de otimização resultam numa simplificação do processo de obtenção dos parâmetros ótimos do modelo. O problema de otimização do classificador LSSVM pode ser descrito como:

$$\text{minimizar} \qquad f_o(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\frac{1}{2}\sum_{i=1}^{N}\xi_i^2 \tag{12}$$

$$\text{s.a.} \qquad y_i[\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) + b] = 1 - \xi_i\ , \quad i = 1, \ldots, N, \tag{13}$$

onde $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ são os pares de vetores de atributos e classes disponíveis durante o treinamento.

Diferentemente do classificador SVM original as funções de restrição são de igualdade em vez de inequações, já na função objetivo as variáveis que representam a margem flexível do modelo, ξ_i, são elevadas ao quadrado, retirando a restrição do modelo SVM tradicional que elas deveriam ser não-negativas.

Resolvendo o problema de otimização pelo seu dual [18] encontramos um sistema de equações lineares, um conjunto de equações de Karush-Khun-Tucker (KKT):

$$\begin{bmatrix} 0 & \mathbf{y}^T \\ \mathbf{y} & \Omega + \gamma^{-1}I \end{bmatrix}\begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}, \tag{14}$$

em que, $\Omega_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{y} = [y_1\ y_2\ \ldots\ y_N]^T$, $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \ldots\ \alpha_N]^T$ e $\mathbf{1} = [1\ 1\ ...\ 1]^T$. A função $K(\mathbf{x}_i, \mathbf{x}_j)$ representa o kernel, ou produto escalar, do mapeamento não-linear dos vetores de atributos $\boldsymbol{\phi}(\mathbf{x}_i)$ e $\boldsymbol{\phi}(\mathbf{x}_j)$. A função discriminante do classificador LSSVM pode ser escrita como

$$f(\mathbf{x}) = sinal\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{15}$$

Uma forma bem conhecida de resolver a equação (14) é através da minimização do quadrado dos erros, que pode ser escrita como o problema de otimização

$$\text{minimizar} \quad f_o(\mathbf{r}) = \frac{1}{2}||A\mathbf{r} - \mathbf{s}||^2. \tag{16}$$

Que possui a solução analítica

$$\mathbf{r} = A^{\dagger}\mathbf{s}, \tag{17}$$

em que $A^{\dagger}$ é a pseudo-inversa de A, definida como

$$A^{\dagger} = (A^T A)^{-1}A^T. \tag{18}$$

Mais detalhes sobre o classificador LSSVM podem ser encontrados em [18].

IV. Proposta: Classificador LSSVM Local

Esta seção é dedicada a explicar minuciosamente o processo de treinamento e predição de modelos de classificação local baseado em protótipos. O processo de treinamento e teste utilizando o classificador LSSVM local está descrito no fluxograma da Figura 1. O fluxograma está dividido nas etapas de treino e teste, que serão detalhadas nas seguintes subseções.

A. Treino

A construção do classificador local depende de duas etapas essenciais: definição das partições e treinamento dos classificadores para cada partição.

1) **Definição das partições**: em termos gerais, é o processo que envolve a definição de quais partições representam os dados. Esta etapa ocorre dentro do processo de treinamento do classificador, em que as partições são definidas e os padrões de entrada são particionados.
2) **Treino por partição**: esta etapa está relacionada treinamento de um modelo para cada subconjunto (partição) dos dados. Portanto, para cada agrupamento definido, um modelo é treinado utilizando apenas os dados do agrupamento.

Na etapa de treino do classificador LSSVM local (L-LSSVM), o conjunto de dados de treino é processado pelo

algoritmo K-médias, o qual definirá agrupamentos ou *clusters*, os quais determinaram as partições locais do classificador L-LSSVM. Note que o hiper-parâmetro K define a quantidade de *clusters* obtido pelo método de agrupamento, onde cada vetor-protótipo do conjunto $\{\mathbf{w}_i\}_{i=1}^{K}$ representa um agrupamento. Em outras palavras, cada agrupamento é composto por um subconjunto da população total de dados de treinamento, e este subconjunto é representado por um vetor-protótipo.

A matriz $\mathbf{W}$ contém os protótipos que representam os *clusters* determinados pelo K-médias. Esta matriz será parte do modelo de classficação local L-LSSVM. Após a definição das K partições locais, formada por subconjuntos do conjunto de dados de treino, os modelos locais serão construídos a partir do classificador LSSVM aplicado em cada uma destas partições. Portanto, o modelo L-LSSVM é formado pela matriz $\mathbf{W}$, responsável pela definição das partições locais; e pelo conjunto de modelos locais $\{M_i(\cdot)\}_{i=1}^{K}$.

B. Teste

No processo de predição utiliza-se o conjunto de teste, que consiste de dados de entrada não-rotulados, os quais são apresentados para o modelo. A predição utilizando o classificador local depende de duas etapas essenciais: seleção da partição e predição local.

1) **Seleção da partição**: é o processo que envolve selecionar a partição local $V_{\mathbf{x}}$ de um novo padrão de entrada $\mathbf{x}$ a partir de alguma medida de similaridade (i.e., distância euclidiana). A partição local que apresenta dados mais similares ao padrão $\mathbf{x}$ será selecionada.
2) **Predição local**: esta etapa está relacionada ao processo de predição da classe a partir de vetores de entrada não rotulados. Após a seleção da partição local do vetor $\mathbf{x}$, denotado por $V_{\mathbf{x}}$, utiliza o modelo $M_{\mathbf{x}}$ como o mais apropriado para prever a classe deste padrão de entrada.

A partição local no classificador L-LSSVM é definida a partir de medidas de similaridade (i.e., distância euclidiana) entre o padrão a ser classificado e os protótipos determinados pelo algoritmo K-médias. O protótipo de menor distância ao padrão não-rotulado $\mathbf{x}$ determinará a partição local a qual $\mathbf{x}$ pertence. O índice do vetor-protótipo de menor distância, denotado por i^*, é obtido a partir da equação

$$i^* = \arg \min_{1 \leq i \leq K} \|\mathbf{w}_i - \mathbf{x}\|_2^2 . \tag{19}$$

Uma vez definido o cluster de índice i^*, utiliza-se o modelo M_{i^*} para realizar a predição da classe referente aquele padrão não-rotulado. Note que cada modelo M_i é um classificador LSSVM construído a partir dos dados da i-ésima partição.

V. Metodologia de Avaliação

A. Conjuntos de dados

Os conjuntos de dados utilizados nos experimentos foram selecionados no UCI Machine Learning Repository [19] e eles são descritos na Tabela I. A segunda e quarta colunas da tabela indica respectivamente o número de classes e atributos para cada conjunto de dados, considerando também suas variações. Portanto, o conjunto de dados *Vertebral Column* possui duas variações, para 2 e 3 classes; o conjunto de dados *Wall-Following* é constituído de três variações considerando 2, 4 e 24 atributos de entrada. Para todos os conjuntos de dados, os atributos de entrada foram normalizados para valores entre 0 e 1.

Tabela I: Descrição dos conjuntos de dados

Dataset	**Classes**	**Nº de Amostras**	**Atributos**
Parkinson	2	195	22
Vertebral Column	$\{2, 3\}$	310	6
Wall-Following	4	5456	$\{2, 4, 24\}$

B. Avaliação de desempenho de classificação

Para avaliar o desempenho de classificação do método estudado, foram utilizadas as seguintes métricas baseadas em matrizes de confusão:

$$accurácia = \frac{TP + TN}{TP + TN + FP + FN}, \tag{20}$$

$$sensibilidade = \frac{TP}{TP + FN}, \tag{21}$$

$$especificidade = \frac{TN}{TN + FP}, \tag{22}$$

onde TP, TN, FP and FN significa *True-Positive*, *True-Negative*, *False-Positive* e *False-Negative*, respectivamente. Adicionalmente, a métrica F_1-Score, descrita na Equação (23), foi adotada como critério para determinar qual modelo apresentou melhor desempenho de classificação:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{23}$$

Para mais detalhes sobre o cálculo e a interpretação das métricas da matriz de confusão, consulte os trabalhos de [20] e [21].

C. Experimentos

Os experimentos foram projetados para comparar o desempenho dos classificadores LSSVM global (G-LSSVM) e LSSVM local (L-LSSVM). Eles foram avaliados usando métricas baseadas em matriz de confusão: *acurácia*, *sensibilidade*, *especificidade* e F_1-Score. Para avaliar estatisticamente os modelos, o método de validação cruzada conhecido por *hold-out* foi aplicado nas simulações. O experimento foi composto por 50 rodadas de separação treinamento/teste de forma estratificada, ou seja, mantendo a proporção original das classes em cada conjunto de treino e teste, em que 50% dos dados foram reservados para treinamento e 50% para teste. É importante observar que em cada rodada de validação cruzada *hold-out*, um novo conjunto de partições é encontrado, pois os dados de treinamento são diferentes em cada execução. Isso elimina qualquer viés na definição de partições, que poderia existir se ela fosse feita usando o conjunto de dados completo.

A otimização dos hiper-parâmetros do modelo envolve dois processos: otimização do número de partições locais e otimização dos hiper-parâmetros do modelo LSSVM.

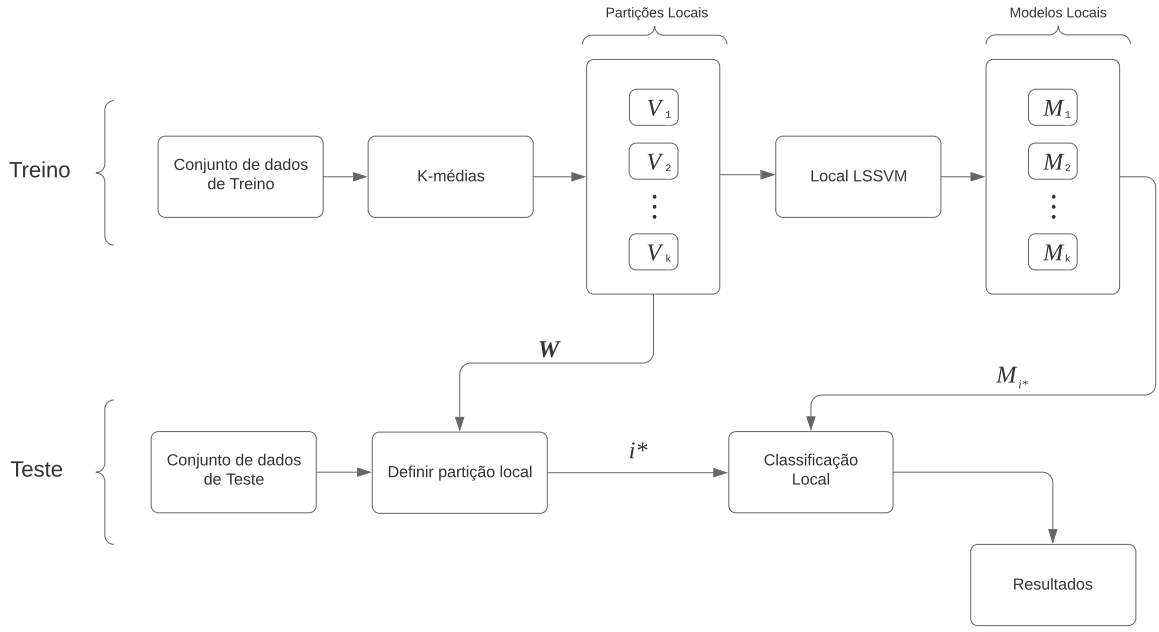


Figura 1: Diagrama que descreve o processo de treino e teste do classificador LSSVM Local

1) Otimização das partições locais: A determinação da quantidade de partições locais foi realizada a partir de um comitê de métricas de validação de agrupamentos (Seção II-C). Para cada rodada de treinamento/teste, utilizou-se os dados de treinamento para estimar o valor ótimo de K que representa a quantidade de partições locais. Neste processo, o algoritmo K-médias foi executado variando o $K \in \{2, 3, \ldots, \lfloor\sqrt{N}\rfloor\}$, onde N representa o número de amostras disponíveis durante o treinamento.

Após a execução do algoritmo de agrupamento para todos esses valores de K os índices de validação foram executados. Dentre as sugestões de valor ótimo de K, fornecida pelos índices, aplicou-se validação cruzada estratificada em 5-fold para determinar o índice que apresenta maior pontuação, considerando a acurácia de classificação e sua estabilidade nos resultados. A pontuação utilizada é dada por

$$score(K) = \mu_{Acc} - 2\sigma_{Acc}, \tag{24}$$

onde μ_{Acc} e σ_{Acc} representam a média e o desvio padrão dos resultados de acurácia, considerando os folds da validação cruzada 5-fold. A utilização desta metodologia baseia-se na premissa de que os índices de validação são adequados para estimar o número ideal de partições, bem como que a utilização de múltiplos índices minimiza a possibilidade de determinar um número inapropriado de partições locais.

2) Otimização dos hiper-parâmetros do classificador LSSVM: Uma busca em grade foi realizada para determinar:

- O valor de σ no kernel gaussiano:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}\right], \tag{25}$$

  onde $\sigma \in \{10^{-0.5}, 10^{-0.375}, 10^{1.25}, 10^{2.125}, 10^{3}\}$;

- Valor do coeficiente γ, que controla a regularização do classificador LSSVM, onde $\gamma \in \{10^{-6}, 10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}\}$.

VI. Resultados e Discussão

Nesta seção, os resultados das simulações com os modelos G-LSSVM e L-LSSVM são apresentados e discutidos. Além disso, a estabilidade de ambos os modelos é avaliada analisando os resultados de dispersão da acurácia referente aos dados de teste entre todas as 50 rodadas. A distribuição referente ao número de *clusters* ou partições para cada rodada também é discutida, pois o desempenho dos classificadores de abordagem local depende diretamente do número de partições consideradas.

As tabelas II, III e IV apresentam os resultados dos classificadores G-LSSVM e L-LSSVM aplicados aos conjuntos de dados *Parkinson*, *Vertebral Column-2C* e *Vertebral Column-3C*, respectivamente. Os resultados do conjunto de dados *Parkinson* mostrou que o modelo L-LSSVM apresenta *sensibilidade* e *especificidade* mais balanceadas. Note que a *especificidade* do L-LSSVM é significativamente maior que a do G-LSSVM. Além disso, em média, o L-LSSVM apresentou maior desempenho de classificação, embora a diferença seja bastante pequena. Em contraste, o conjunto de dados *Vertebral Column-2C* e *Vertebral Column-3C* apresentaram resultados em favor do modelo G-LSSVM, em termos de desempenho médio de classificação. Note que a diferença é sutil e pouco conclusiva.

Tabela II: Desempenho do conjunto *Parkinson*

	Modelo	Acurácia	Sens.	Espec.	F_1
Treino	G-LSSVM	97.34 ± 4.16	99.67	90.25	98.32
	L-LSSVM	**98.70 ± 2.53**	**99.73**	**95.58**	**99.16**
Teste	G-LSSVM	87.02 ± 3.73	95.49	60.92	91.72
	L-LSSVM	**88.31 ± 4.14**	**94.19**	**70.17**	**92.39**

Tabela III: Desempenho do conjunto *Vertebral Column*-2C

	Modelo	Acurácia	Sens.	Espec.	F_1
Treino	**G-LSSVM**	**89.32 ± 3.51**	**81.04**	**93.26**	**82.88**
	L-LSSVM	87.92 ± 4.76	78.00	92.65	80.49
Teste	**G-LSSVM**	**82.98 ± 2.69**	**70.52**	**88.91**	**72.53**
	L-LSSVM	82.01 ± 3.72	69.60	87.92	71.36

Tabela IV: Desempenho do conjunto *Vertebral Column*-3C

	Modelo	Acurácia	Sens.	Espec.	F_1
Treino	G-LSSVM	86.86 ± 3.53	93.21	82.45	93.24
	L-LSSVM	**88.34 ± 3.99**	**93.86**	**84.61**	**94.02**
Teste	**G-LSSVM**	**83.19 ± 3.33**	**91.00**	**78.52**	**90.95**
	L-LSSVM	81.35 ± 3.35	89.77	76.55	89.83

Tabela V: Desempenho do conjunto *Wall-Following*-2a

	Modelo	Acurácia	Sens.	Espec.	F_1
Treino	G-LSSVM	96.44 ± 0.22	98.68	95.76	98.65
	L-LSSVM	**98.47 ± 0.39**	**99.44**	**98.45**	**99.43**
Teste	G-LSSVM	96.15 ± 0.39	98.56	95.06	98.54
	L-LSSVM	**97.80 ± 0.45**	**99.19**	**97.34**	**99.19**

Tabela VI: Desempenho do conjunto *Wall-Following*-4a

	Modelo	Acurácia	Sens.	Espec.	F_1
Treino	G-LSSVM	97.20 ± 0.48	98.94	97.27	98.92
	L-LSSVM	**97.79 ± 0.73**	**99.16**	**97.90**	**99.15**
Teste	**G-LSSVM**	**96.19 ± 0.42**	**98.54**	**95.64**	**98.53**
	L-LSSVM	96.09 ± 0.42	98.50	95.55	98.50

Tabela VII: Desempenho do conjunto *Wall-Following*-24a

	Modelo	Acurácia	Sens.	Espec.	F_1
Treino	G-LSSVM	97.18 ± 0.52	98.89	96.78	98.92
	L-LSSVM	**97.53 ± 0.39**	**99.03**	**97.18**	**99.06**
Teste	**G-LSSVM**	**91.10 ± 0.52**	**96.42**	**89.84**	**96.47**
	L-LSSVM	90.60 ± 0.71	96.21	89.42	96.26

As tabelas V, VI e VII descrevem os resultados dos classificadores aplicados ao conjunto de dados *Wall-Following*, considerando 2, 4 e 24 atributos no padrão de entrada. Os resultados referentes ao conjunto de dados *Wall-Following* mostrou que o modelo L-LSSVM e G-LSSVM apresentam resultados equivalentes de desempenho de classificação.

Com relação ao *Wall-Following-2a*, o método L-LSSVM apresentou maior desempenho médio. Podemos notar que ambos classificadores L-LSSVM e G-LSSVM apresentaram resultados bastante próximos em termos médios de desempenho de classificação. Os resultados apresentados sugerem que os modelos locais apresentam menor vantagem quando combinado com classificadores não-lineares (i.e., LSSVM). Em outras palavras, se o problema de classificação é solucionável com um classificador global com solução não-linear, é preciso avaliar de fato a vantagem de se utilizar o modelo local.

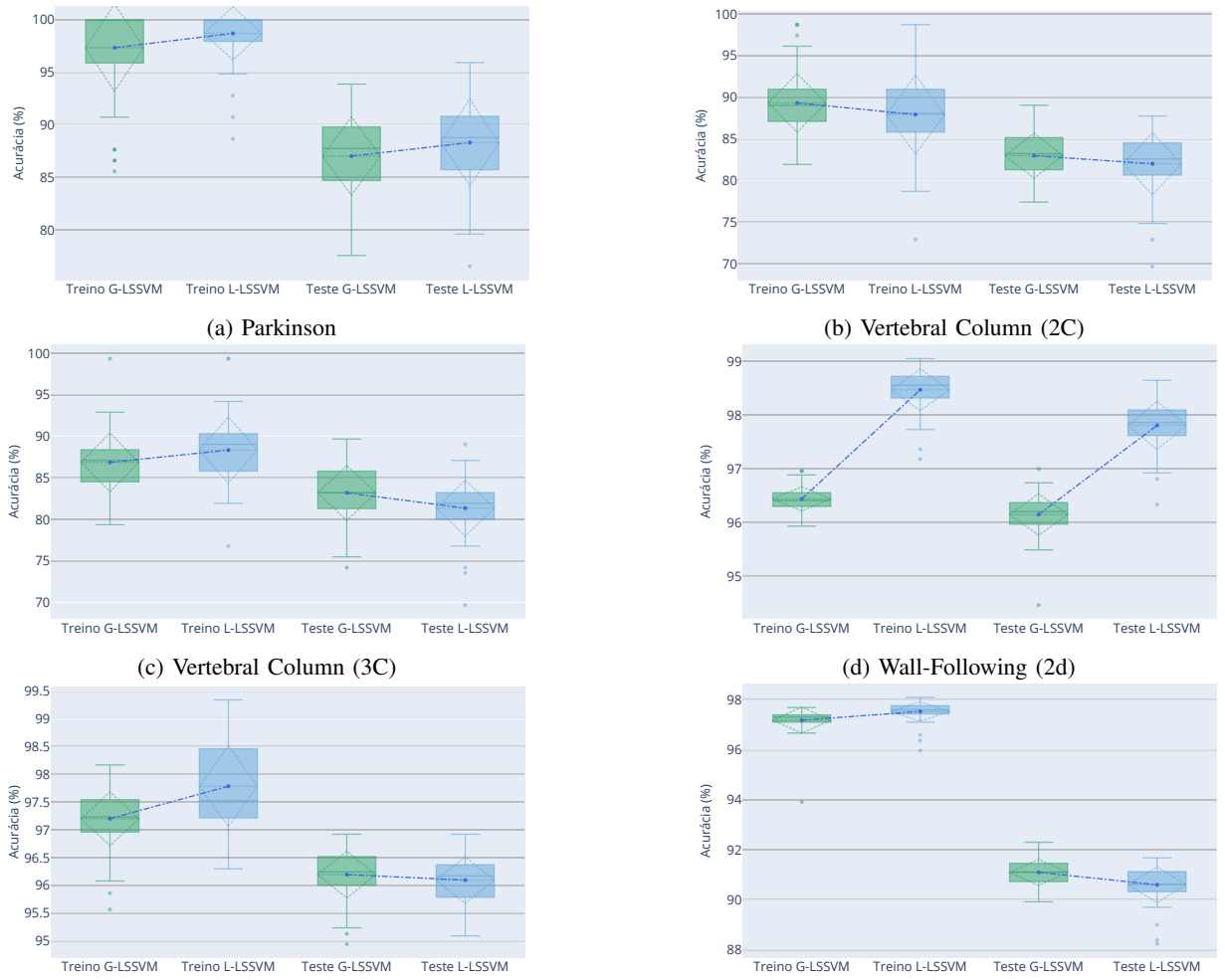
O conjunto de dados *Wall-Following* consiste em dados desbalanceados e com alta sobreposição entre instâncias de diferentes classes. Portanto, isto sugere que um classificador não-linear será mais apropriado para solução deste problema; No contexto dos modelos locais estudado neste trabalho, conjuntos de dados como este apresentam uma tendência que índices de validação façam sugestões de valores variados para K. Portanto, é preciso garantir que as partições locais sejam definidas de tal forma que os dados presentes em cada partição sejam suficientes para construção do modelo local. No caso do modelo local, a presença de várias partições pode prejudicar o desempenho de classificação do modelo, uma vez que partições podem conter poucos dados de uma determinada classe. Conjuntos de dados desbalanceados são mais sujeitos a essa problemática.

A avaliação do modelo a partir da distribuição dos resultados de desempenho médio de classificação (i.e., acurácia) revela informações relacionadas estabilidade do modelo. A Figura 2 ilustra as distribuições de acurácia envolvendo as 50 rodadas do experimento de validação cruzada *hold-out*.

Os diagramas de caixa (*boxplots*) ilustrados na Figura 2 mostram que ambos os classificadores G-LSSVM e L-LSSVM apresentam distribuições similares em desempenho de classificação, considerando o conjunto de dados de teste. Em geral, o classificador L-LSSVM apresenta em quase todos os conjunto de dados maior desempenho de classificação nos dados de treino, o que sugere uma tendência a *overfitting*. É importante destacar que essa tendência a *overfitting* pode ser resultado da má definição das partições locais. Em geral, algoritmos baseados em modelos locais tendem a apresentar maior dispersão nos resultados devido sua dependência em relação ao hiper-parâmetro K, responsável por definir a quantidade de partições locais. É importante observar que a definição das partições locais é essencial para garantir que o classificador L-LSSVM tenha alto desempenho de classificação. Cada rodada do experimento pode resultar em diferentes valores de K^*. Logo, se o treinamento do K-médias resultar em partições locais mal condicionadas (desbalanceadas ou com poucas amostras para treino), isto afetará diretamente o desempenho do classificador local.

A Figura 3 mostra o histograma do número de partições locais identificadas por rodada referente ao processo de validação cruzada *hold-out*. É possível verificar que o número de partições varia muito em rodadas diferentes, pois o algoritmo K-médias processa conjuntos de treinamento distintos, o que resulta em seleção de diferentes conjuntos de partições.

Esta variabilidade no número de partições definidas depende diretamente da organização dos dados no espaço de entrada.



(a) Parkinson

(b) Vertebral Column (2C)

(c) Vertebral Column (3C)

(d) Wall-Following (2d)

(e) Wall-Following (4d)

(f) Wall-Following (24d)

Figura 2: Distribuição dos resultados de acurácia do experimento utilizando validação cruzada *hold-out*

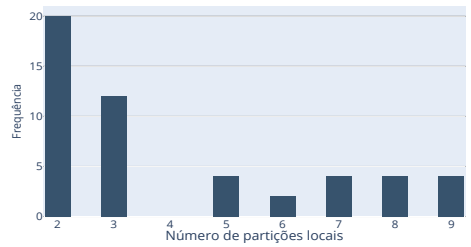
Podemos verificar na Figura 3 que quanto maior a quantidade de amostras, maior será a tendência de sugestões de valores altos para K^*. As Figuras 2 (a), 2 (b) e 2 (c), apresentaram com maior frequência valores de $K^* = 2$. Note que, com relação ao *Wall-Following*, ambas figuras 2 (d), 2 (e) apresentaram valores mais dispersos, com exceção da Figura 2 (f).

VII. Conclusão e Trabalhos Futuros

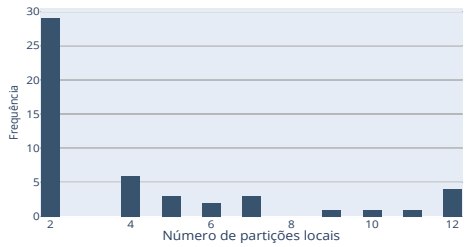
Neste artigo, uma estrutura de classificação local foi apresentada e avaliada. A estrutura de classificação local baseia-se na combinação algoritmo de agrupamento K-médias e de classificadores LSSVM. Utilizando este classificador de abordagem local, comparamos e realizamos uma avaliação abrangente do desempenho deste modelo local de classificação e o modelo global do classificador LSSVM, aplicados a três conjuntos de dados de referência. Os resultados obtidos mostraram que a abordagem de classificação local utilizada apresenta-se como uma ferramenta promissora na solução de problemas de classificação não linearmente separáveis. Os resultados mostram que o método proposto tende a equilibrar o desempenho da classificação para todas as classes, além de garantir resultados mais significativos com conjuntos de dados que apresentam muitas amostras em partições com sobreposição entre classes. Observou-se também que conjuntos de dados desbalanceados tendem a causar um viés na superfície de separação, de forma que a classe com mais amostras seja priorizada em relação às outras classes. Atualmente, os autores estão trabalhando em métodos que otimizam o processo de determinação das partições locais, uma vez que desempenha um papel essencial no desempenho da classificação do modelo local. Como sugestão de trabalho futuro é o desenvolvimento de um índice específico para determinação das partições locais, bem como o uso de outras técnicas de quantização vetorial e agrupamento para determinação das partições locais.

Agradecimentos

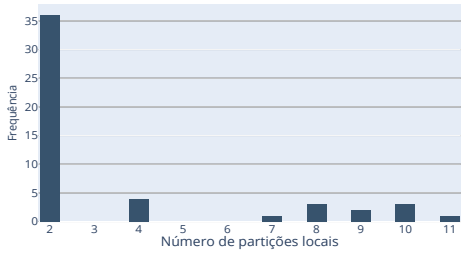
O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores também agradecem à FUNCAP (Concessão no. 88887.177150/2018-00) e ao CNPq (Concessão no. 309451/2015-9) por apoiarem esta pesquisa.



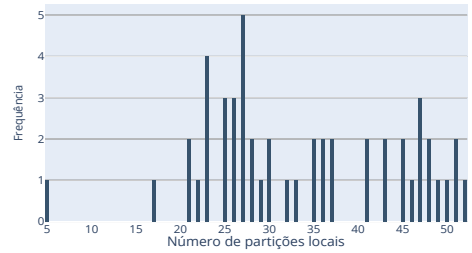
(a) Parkinson



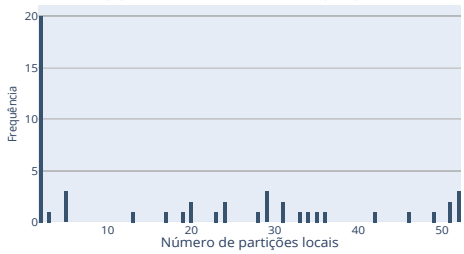
(b) Vertebral Column (2C)



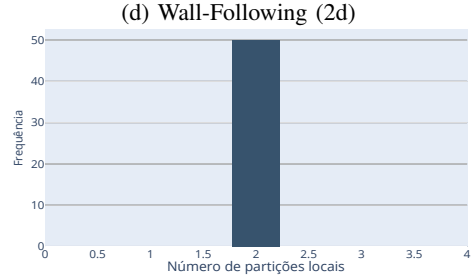
(c) Vertebral Column (3C)



(d) Wall-Following (2d)



(e) Wall-Following (4d)



(f) Wall-Following (24d)

Figura 3: Histograma do número de partições locais envolvendo o experimento de validação cruzada *hold-out*

REFERÊNCIAS

[1] X. Wang and V. L. Syrmos, "Nonlinear system identification and fault detection using hierarchical clustering analysis and local linear models," in *2007 Mediterranean Conference on Control & Automation*. IEEE, 2007, pp. 1–6.

[2] A. H. S. Júnior, G. A. Barreto, and F. Corona, "Regional models: A new approach for nonlinear system identification via clustering of the self-organizing map," *Neurocomputing*, vol. 147, pp. 31 – 46, 2015, advances in Self-Organizing Maps Subtitle of the special issue: Selected Papers from the Workshop on Self-Organizing Maps 2012 (WSOM 2012).

[3] M. D. Marzio, S. Fensore, A. Panzera, and C. C. Taylor, "Local binary regression with spherical predictors," *Statistics & Probability Letters*, vol. 144, pp. 30 – 36, 2019, advances in statistical methods and applications for Climate change and Environment.

[4] E. Alpaydin and M. I. Jordan, "Local linear perceptrons for classification," *IEEE Transactions on Neural Networks*, vol. 7, no. 3, pp. 788–794, May 1996.

[5] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991.

[6] M. Bevilacqua and F. Marini, "Local classification: Locally weighted–partial least squares-discriminant analysis (lw–pls-da)," *Analytica Chimica Acta*, vol. 838, pp. 20 – 30, 2014.

[7] W. Song, H. Wang, P. Maguire, and O. Nibouche, "Local partial least square classifier in high dimensionality classification," *Neurocomputing*, vol. 234, pp. 126 – 136, 2017.

[8] J. Peng and B. Bhanu, "Local discriminative learning for pattern recognition," *Pattern Recognition*, vol. 34, no. 1, pp. 139 – 150, 2001.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[10] Y. Yusof and Z. Mustaffa, "A review on optimization of least squares support vector machine for time series forecasting," *Int J Artif Intell & Applications*, vol. 7, no. 2, pp. 35–49, 2016.

[11] A. Shokrollahi, M. Arabloo, F. Gharagheizi, and A. H. Mohammadi, "Intelligent model for prediction of co2–reservoir oil minimum miscibility pressure," *Fuel*, vol. 112, pp. 375–384, 2013.

[12] S. N. Jothiraj, T. G. Selvaraj, B. Ramasamy, N. P. Deivendran, and M. Subathra, "Classification of eeg signals for detection of epileptic seizure activities based on feature extraction from brain maps using image processing algorithms," *IET Image Processing*, vol. 12, no. 12, pp. 2153–2162, 2018.

[13] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, July 2002.

[14] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[15] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[16] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[17] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[18] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[19] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[20] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340-341, pp. 250 – 261, 2016.

[21] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427 – 437, 2009.